# Generic Database Design for Patient Management Information

Stephen B. Johnson, Ph.D [1], Terre Paul [2], Anna Khenina [1]

[1] Department of Medical Informatics
Columbia University, New York

[2] Clinical Information Services
Presbyterian Hospital, New York

*Patient management information tracks general facts about the location of the patient and the providers assigned to care for the patient. The Clinical Data Repository at Columbia Presbyterian Medical Center employs a generic schema to record patient management events. The schema is extremely simple, yet can support several different views of patient information, as required by different applications: a longitudinal view of patient visits, including both inpatient and outpatient encounters; a visit-oriented view, to record facts related to a current encounter; a location-based view to provide a census of a nursing ward; and a provider-based view to give a list of the patients currently being cared for by a given clinician. All of these views can be supported in a highly efficient manner by the use of appropriate indexes.*

## INTRODUCTION

A clinical data repository is central resource that integrates data from many different computer applications. Traditionally, such databases have concentrated on the diagnostic and therapeutic services performed on behalf of patients. Some systems also manage the orders which request these services. Clinical assessments of a patient's condition are also collected in the form of problem lists, admitting diagnoses, and discharge diagnoses.

In some clinical information systems, the representation of a patient encounter is implicit, and must be inferred from the existence of diagnostic results. Other systems model either an explicit outpatient visit or inpatient stay, but often cannot represent both in the same database. Most administrative and financial systems only approximate patient encounters in the form of a case or account. The majority of systems do not have a general way of representing similar administrative events, such as recording a patient's primary physician, or tracking the assignment of residents to an inpatient.

An alternative approach to these designs is to model a general class of clinical events called "patient management events", which include admissions, discharges, transfers, clinic visits, assignment of physicians, and similar events. A number of examples are classified on the left side of Table 1.

This paper present a generic database design for patient management events. This approach enables a clinical repository to integrate encounter information from both inpatient and outpatient settings. It provides a method for modeling new types of events that become important as applications evolve, e.g., the residents assigned to a patient, or visiting nurse encounters in the patient's home.

The design for patient management events has been implemented in the Clinical Data Repository [1-3] at Columbia-Presbyterian Medical Center (CPMC), a database containing clinical information for 1.3 million patients, covering a period of roughly six years. At the present time the bulk of the data in the repository consists of laboratory results, medication orders, radiology procedures, demographic data, and transcribed reports (radiology, pathology, cardiology, discharge summaries, etc.).

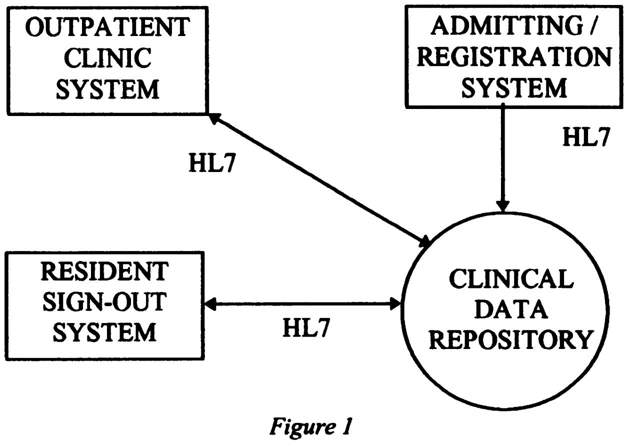

*Figure 1*

1091—8280/97/$5.00 © 1997 AMIA, Inc.

| EVENT TYPE | ATTRIBUTES |
|---|---|
| patient management event | patient id, primary time, organization, provider, event status |
| inpatient management event | location |
| admission | stop time, room, bed, referring physician, admit source |
| discharge | patient disposition |
| transfer | transport method, room, bed |
| outpatient encounter | location, encounter note |
| emergency room visit | |
| doctor private office visit | |
| clinic visit | billing diagnosis, billing procedure |
| ambulatory surgery visit | |
| patient home encounter | |
| visiting nurse service visit | |
| provider assignment | coverage comment |
| assignment of primary physician | |
| assignment of primary resident | |
| assignment of secondary resident | |

*Table 1- Patient management events and their attributes*

Patient management events are gathered by many different applications in an information system. At the current time, CPMC has three such applications, the admitting system which captures inpatient and outpatient encounters, an outpatient system used to capture information about clinic visits, and a system which aids residents in "signing out" their patients to the next resident on call. Each of these applications communicate with the Clinical Data Repository using the Health Level 7 (HL7) protocol, as depicted in Figure 1.

Each clinical application has different data requirements, which can evolve over time. If the repository schema models application data too directly, then the schema must undergo constant changes, as applications add new data elements, and change the characteristics of existing ones[4]. Database management systems allow the schema to be altered dynamically, but this creates a maintenance problem for system administrators, and will, over time lead to significant performance problems[5]. An alternative approach is to use a generic database design, which captures the common structure of clinical information that enables applications to share data, while also allowing applications to evolve in a smooth manner[6-10].

Clinical applications also require efficient access to patient data to create new information, update existing information, and to query for information created by other applications. Patient management events are somewhat more complicated than other events in the repository, in terms of the number of different views of the patient record that must be supported efficiently:

- patient-oriented: longitudinal view of patient encounters
- visit-oriented: access to a given admission or outpatient visit to query or update information
- location-oriented: determining which patients are present at a given location
- provider-oriented: determining which patients are currently assigned to a given provider.

In general, the clinical repository is designed to support users involved in patient care (physicians, nurses, administrators, etc.), not researchers, and thus is optimized for queries for a single patient. Note, however, that the last two views in the above list are cross-patient views that must be supported efficiently.

## METHODS

Data requirements were gathered for three applications: an inpatient registration system, an outpatient clinic system, and an inpatient resident sign-out system (Figure 1). This resulted in the categories of events as listed in Table 1. The attributes common to all patient management events are listed in the right column of the first row: the patient involved in the event (represented as medical record number), the primary date and time when the event occurred (or began), the organization responsible (identifier of a department or institution), and the principal clinical provider. Since the database tracks events over time, the event status attribute is

| MRN | PRIMARY TIME | ORG | EVENT ID | EVENT CODE | EVENT STATUS | LOC | PROVIDER | EVENT LINK |
|---|---|---|---|---|---|---|---|---|
| 12345 | 100595 0930 | PH | A55501 | 32467 | I | M6GN | D9927 | -- |
| 12345 | 100795 1100 | PH | A55502 | 32468 | I | M5GS | D9927 | A55501 |
| 12345 | 100995 1630 | PH | A55503 | 32469 | I | M5GS | D9927 | A55501 |
| 12345 | 062196 1300 | ACNC | C22200 | 47638 | I | VC2 | D3321 | -- |
| 12345 | 031797 0800 | PH | A77701 | 32350 | A | M4HS | D0707 | -- |

*Table 2- Patient management events*

require to specify whether an event is currently active or not. For example, a current admission has an event status of "active", but a prior admission would be "inactive".

Some examples of additional attributes are listed for some of the other events in the table. Inpatient events require a nursing station location, while outpatient events must record a location for the particular emergency room, clinic, etc. Admissions and transfers refine location information further by specifying a room and bed. Some events have additional roles that providers can play, e.g. the referring physician for an admission event. The duration of most patient management events is not clinically relevant, but for those events that require this information (e.g. admissions), the "stop time" attribute is used. While most of these events communicate only limited clinical information, a few specify some clinical attributes, e.g., the disposition of the patient at discharge, or the note summarizing a clinic visit.

The hierarchy of patient management events in Table 1 could be implemented in a relational database using many tables with many specific columns. An alternative approach is to construct an extremely simple database design consisting of just two tables. This is accomplished in two steps, using the techniques of generic data modeling [6].

In the first step, the whole hierarchy is collapsed into a single table that contains the union of all the attributes of all of the subtypes. A special attribute called "event code" must be added to this table in order to distinguish which of the several different event types is intended for a given row of the database. To do this, each patient management event is assigned a unique code, using the CPMC Medical Entities Dictionary. For example, to record an admission in this table, the event code column is given the value 32467, while a clinic visit would have the value 47638.

This single table database design allows all of the patient management events listed in Table 1 to be stored. However, because there is a specific column for each distinct attribute, the design wastes space, and is very inflexible. For any given row of the table, many of the columns will not be used. For example, assignment of providers does not require recording a room and a bed. A more serious problem is that every time an application requires a new attribute to be stored, another column must be added to the table. For example, there are many different roles that providers may play other than being the referring physician, so new columns would be required for each distinct role. As noted above, database management systems allow addition of new columns, but query performance will degrade over time due to fragmentation of data in the table.

The second step of generic data modeling addresses these issues of flexibility by creating a second table which is used to store additional attributes associated with a given patient management event. The attributes that are considered to be important to all patient management events are kept in the main table (Table 2).

| MRN | PRIMARY TIME | COMP-ONENT | COMP CODE | VALUE |
|---|---|---|---|---|
| 12345 | 100595 0930 | 1 | 48489 | D3322 |
| 12345 | 100595 0930 | 2 | 50027 | ER01 |
| 12345 | 100995 1630 | 1 | 50031 | wheel chair |
| 12345 | 062196 1300 | 1 | 40375 | 466.0 |
| 12345 | 062196 1300 | 2 | 40376 | 531 |

*Table 3- Patient management event components*

All the remaining attributes are assigned codes in the Medical Entities Dictionary, and stored in a second table called the "component table" (Table 3).

The attributes selected to remain in the main table are those which:

- apply to the majority of event types,
- have only a single value for a given event,
- are useful for indexing for efficient retrieval.

The columns chosen for the main table of the CPMC Clinical Data Repository (Table 2), record the patient identifier (MRN), the date and time of the event (PRIMARY TIME) the organization responsible (ORG), the identifier used by the application storing the data (EVENT ID), the dictionary code used to specify the type of event (EVENT CODE), the active/inactive status of the event (EVENT STATUS), the location within the institution of the event (LOCATION), and the provider who has primary responsibility for this event (PROVIDER). It is also useful to allow one management event to refer to another, e.g. a discharge event must refer to the admission which it is terminating. This relationship is modeled using the EVENT LINK column, which contains the unique identifier of the event being referred to.

The component table (Table 3) is used to store all the other attributes associated with a patient management event. There can be many attributes for a given event. The medical record number and primary time of the event are stored for each attribute, and the attributes are numbered consecutively using the COMPONENT column. Each attribute stored in this table has a unique code defined in the Medical Entities Dictionary, which is stored in the COMP CODE column. The VALUE column is used to store the value associated with a given attribute. This column is a varying length character column, which can store a wide variety of data formats. For example, to record the referring physician of an admission, the code 48489 is stored in COMP CODE, and the identifier of the physician is stored in the VALUE column.

This design is extremely flexible. Whenever a new attribute is required, it is only necessary to define a new code for the attribute in the dictionary. New information about a patient can then immediately be stored in the database, without having to alter the database schema.

Applications also require highly efficient access to the clinical repository. In a relational database, efficient access is achieved using indexes. Thus, it is essential that the attributes used to locate information in the database are placed in the event table (Table 2), where they can be indexed. As described above in the introduction, applications require efficient access to patient management events from four points of view: patient, event, location, and provider. Efficient access can be achieved by creating an index for each of these views, as shown in Table 4.

| **INDEX** | **COLUMNS** | | | |
|---|---|---|---|---|
| patient | MRN | PRIMARY TIME | -- | -- |
| event | ORG | EVENT ID | -- | -- |
| location | LOCATION | EVENT STATUS | EVENT CODE | MRN |
| provider | PROVIDER | EVENT STATUS | EVENT CODE | MRN |

*Table 4*
*Indexes on patient management events*

The "patient" index can be used to locate a specific event for a patient using MRN, to provide a complete history of management events given a patient, or focus on a time period by restricting PRIMARY TIME. The "event" index is required by applications that create patient management events. The EVENT ID column is used to find an event (e.g., an admission) using the identifier known to the sending system (e.g., the case number of the admitting system). Since different applications may accidentally use the same identifiers for events, the ORG column is combined with EVENT-ID (in this model, each organization has its own application).

The "location" index is used to determine which patients are present at a given nursing station. Since applications are frequently interested only in the current location for a patient, LOCATION is combined with EVENT-STATUS, which indicates whether the relationship is active (currently valid) or not. The EVENT CODE column is required to restrict the queries to a specific type of event (e.g., admissions). Often, an application only needs to retrieve a patient identifier, so the MRN column is added to the index for efficiency. The "provider" table is used to obtain the list of patients associated with a given provider. This index has the same design as the "location index", except the first column is PROVIDER.

## RESULTS

The two tables and four indexes described above were created as part of the Clinical Data Repository, which is implemented using the DB2 database management system on an IBM mainframe. These tables are used by the three applications shown in Figure 1. The Repository also contains tables for patients, providers, orders, assessments and services performed on behalf of the patient. The database design for management events can be evaluated in terms of the complexity of the programs that access the database, and the performance of these queries.

The outpatient clinic system uses three different SQL queries to update patient management information, and three different queries to retrieve a patient visit and related attribute information. The resident sign-out system uses two different SQL statements to update the database, and two to retrieve information about the residents that care for an inpatient. The admitting system issues twelve distinct types of transaction pertaining to registrations, admissions, discharges and transfers, which are supported by a program using 20 different SQL statements. All queries are very simple and involve at most one relational join.

A test set of 2000 transactions was executed against the present database schema and indexes. The average transaction processing time was 0.073 seconds.

## CONCLUSION

Patient management events constitute an important class of clinical events. When management events are integrated into a database design with other clinical events (orders, services, assessments), they can serve as temporal reference points, helping to form a view of the patient record based on encounters. Patient management events share certain attributes (patient, organization, location, provider), thus it is natural to group them together in the database. Efficient access to patient data can be achieved by creating indexes on these common attributes.

Different events require specialized attributes, which can be accommodated using a generic database design with two tables. This design enables new attributes to be created as necessary, allowing new applications to be incorporated into the clinical information system with minimal impact.

## REFERENCES

1. Johnson SB, Friedman C, Cimino JJ, Clark T, Hripcsak G, Clayton PD. Conceptual Data Model for a Central Patient Database. Proceedings of the Fifteenth Symposium on Computer Applications in Medical Care. Washington, D.C.; 1991: 381-385.

2. Friedman C, Hripcsak G, Johnson SB, Cimino JJ, Clayton PD. A Generalized Relational Scheme for an Integrated Clinical Patient Database. Proceedings of the Fourteenth Annual Symposium on Computer Applications in Medical Care. Washington, D.C. 1990.

3. Johnson SB, Hripcsak G, Chen J, Clayton PD. Accessing the Columbia Clinical Repository. Proceedings of the Eighteenth Annual Symposium on Computer Applications in Medical Care, 1994 November 5-9; Washington (DC). New York: McGraw Hill, 1994.

4. Sjoberg D. Quantifying Schema Evolution. Information and Software Technology, 1993;13(1):35-44.

5. Date C. Introduction to Database Systems, Sixth Edition. New York: Addison-Wesley, 1995.

6. Johnson SB. Generic Data Modeling for Clinical Repositories. JAMIA, 1996:3(5).

7. Barrows R, Johnson SB. A Data Model that Captures Clinical Reasoning about Patient Problems. Proceedings of the Nineteenth Annual Symposium on Computer Applications in Medical Care, 1995 October 28 - November 1; New Orleans (LA). Philadelphia: Hanley and Belfus, 1995:402-5.

8. Dolin RH. A high-level object-oriented model for representing relationships in an electronic medical record. In: Ozbolt J, ed. Eighteenth Symposium on Computer Applications in Medical Care. Philadelphia: Hanley and Belfus, 1994:514-518.

9. Van Ginneken AM, Stam H, Duiterhout JS. A Powerful macro-model for the computer patient record. In: Ozbolt J, ed. Eighteenth Symposium on Computer Applications in Medical Care. Philadelphia: Hanley and Belfus, 1994:496-500.

10. Essin DJ, Lincoln TL. Implementing a Low-cost Computer-based Patient Record: A controlled vocabulary reduces database design complexity. In: Gardner R, ed. Nineteenth Symposium on Computer Applications in Medical Care. Philadelphia: Hanley and Belfus, 1995:431-5.